

# Table of Contents

**Document Engine Example** ..... 2

[Programmers Reference](#), [IDocumentDataSource Interface](#), [Example](#)

## Document Engine Example

The code snippet below represents essentially the full text of the document engine that handles the text document type in **Aware IM**:

Client side component

```
public class TextEngineClient implements IDocEngineClient
{
    public TextEngineClient()
    {
    }
    /**
    *@see IDocEngineClient#getTemplateType()
    */
    public String getTemplateType()
    {
        return CommonConstants.TEMPLATE_TEXT;
    }
    /**
    *@see IDocEngineClient#getEditCommand(String)
    */
    public FileToEdit getEditCommand(String templateFilePath)
    {
        return Utils.isWindows()?
            new FileToEdit("cmd.exe /k", "\"" + templateFilePath + "\""):
            new FileToEdit("", templateFilePath);
    }
    /**
    *@see IDocEngineClient#getTemplateFileType()
    */
    public String getTemplateFileType()
    {
        return "txt";
    }
    /**
    *@see IDocEngineClient#canHaveExternalResources()
    */
    public boolean canHaveExternalResources()
    {
        return false ;
    }
    /**
    * @see
    IDocEngineClient#initialiseTemplate(DataPresentationTemplate, IDomainVersion)
    */
    public void
```

```

initialiseTemplate(DataPresentationTemplatetdef, IDomainVersiondomainVersion)
    throws Exception
{
    // determine attribute paths used
    Stringtext=getTemplateText(tdef);
    if (text!= null )
    {
        // find if there are any tags
        VectortaggedParams=Utils.extractParameters(text,"<<", ">>");
        if (taggedParams!= null &&taggedParams.size()>0)
            DocumentEngineUtils.setAttributePaths(taggedParams,tdef);
    }
}
/**
 * @see
com.bas.shared.docengines.IDocEngineClient#elementsRenamed(com.bas.shared.do
main.configuration.elements.DataPresentationTemplate,com.bas.shared.domain.c
onfiguration.misc.RenamedElements,com.bas.shared.ruleparser.INodeChangeHelpe
r,IDomainVersion)
 */
public DataPresentationTemplateelementsRenamed(
    DataPresentationTemplateoldTemplate,
    RenamedElementsrenamedElements,
    INodeChangeHelperhelper,
    IDomainVersiondomainVersion)
{
    Stringtext=getTemplateText(oldTemplate);
    if (text== null )
        return null ;
        // extract all parameters, convert them to ASTTagStatement
        // expressions, apply changes
        // and then replace parameters in the document
    VectortaggedParams=Utils.extractParameters(text,"<<",">>");
    if (taggedParams== null ||taggedParams.size()==0)
        return null ;
    Vectormodified=DocumentEngineUtils.getModifiedParameters(taggedParams,
renamedElements,helper);
    VectorwithTags=DocumentEngineUtils.appendTags(modified,"<<", ">>");
    StringnewText=Utils.replaceParameters (text,withTags,"<<", ">>");
    if (newText.equals(text))
        return null ;
    DataPresentationTemplateneNewTemplate =
(DataPresentationTemplate)oldTemplate.deepCopy();
    try
    {
        DocumentEngineUtils.setAttributePaths(modified, newTemplate);
        BinaryResourcebr= new BinaryResource
        (oldTemplate.getTemplate().getName(),newText.getBytes());
        newTemplate.setTemplate(br);
    }
    catch (Exception e)

```

```

    {
        return null ;
    } // shouldn't happen
    return newTemplate;
}
protected StringgetTemplateText(DataPresentationTemplatetemplate)
{
    try
    {
        BinaryResourcetemplData=template.getTemplate();
        if (templData== null )
            return null ;
        byte []data=templData.getData();
        if (data== null )
            return null ;
        return new String(data);
    }
    catch (ResourceUnavailableExceptionre)
    {
        return null ;
    }
}
}

```

*// Server side component*

```

public class TextDocumentEngine implements IDocumentEngine
{
    public TextDocumentEngine()
    {
    }
    /**
    *@see IDocumentEngine#getClientEngine()
    */
    public IDocEngineClientgetClientEngine()
    {
        return new TextEngineClient();
    }
    /**
    *@see
    IDocumentEngine#fillWithData(String,Map,IDocumentDataSource,DataPresentation
    Template,IDocEngineHelper,IDomainVersion,int,boolean)
    */
    public BinaryResource[]fillWithData(Stringpath, Mapparams,
    IDocumentDataSourcedataSource, DataPresentationTemplatetdef,
    IDocEngineHelperhelper, DomainVersiondomainVersion, int outputFormat,
    boolean printResults)
        throws DocumentException
    {
        BinaryResourcetemplate=tdef.getLoadedTemplate();
        if (template== null )
            throw new DocumentException("No data in the presentation

```

```

template");
StringfileName=path+"/"+template.getName();
try
{
    FileinputFile= new File(fileName);
    Stringtext=FileUtils.readTextFile(fileName);
    // find if there are any tags
    VectortaggedParams= Utils.extractParameters(text);
    if (taggedParams== null || taggedParams.size()==0 ||
dataSource== null )
    {
        // just return the original template
        BinaryResource[]results= new BinaryResource[1];
        results[0]= new BinaryResource();
        results[0].setName(inputFile.getName());
        results[0].setData(FileUtils.readFile (fileName));
        return results;
    }
    if (dataSource== null )
        throw new DocumentException("Document template requires
data, but the data source is not provided");
        // prepare the data source
        dataSource.reset();
        // for each group of records create resulting document
        // and resolve tagged fields insidethe text
        Vectorresources= new Vector();
        while (dataSource.next())
        {
            VectortagValues= new Vector
            (taggedParams.size());
            for (Iteratoriter=taggedParams.iterator();
                iter.hasNext();)
            {
                StringtagContents=(String)iter.next();
                Object o=dataSource.getExpressionValue (tagContents);
                tagValues.add(o== null ? null :o.toString());
            }
            StringnewText= Utils.replaceParameters(text, tagValues);
            if (printResults)
                printText(newText);
            BinaryResourcebr= new BinaryResource();
            br.setData(newText.getBytes());
            br.setName(inputFile.getName());
            resources.add(br);
        }
        if (resources.size()==0)
            return null ;
        BinaryResource[]results= new BinaryResource [resources.size()];
        int i=0;
        for (Iteratoriter=resources.iterator();iter.hasNext();
            ++i) results[i]=(BinaryResource)iter.next();

```

```
        return results;
    }
    catch (DocumentException de)
    {
        throw de;
    }
    /**
    *@see IDocumentEngine#printFile()
    */
    public void printFile(File fileToPrint)
    throws DocumentException
    {
        try
        {
            String text = FileUtils.readFileToString(fileToPrint.getPath());
            printText(text);
        }
        catch (Exception e)
        {
            throw new DocumentException(e.getMessage());
        }
    }
    protected void printText(String text)
    throws DocumentException
    {
        try
        {
            PlainDocument doc = new PlainDocument();
            doc.insertString(0, text, null);
            TextDocumentPrintRenderer printer = new
TextDocumentPrintRenderer();
            printer.print(doc, false);
        }
        catch (Exception e)
        {
            throw new DocumentException(e.getMessage());
        }
    }
}
}
```

From:  
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:  
<http://www.awareim.com/dokuwiki/docs/3500/0600/0650?rev=1680756400>

Last update: **2023/04/06 04:46**

